НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“___”_____2020 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки : 113 «Прикладна математика»
(код і назва)

на тему: Атака подвійної витрати на протокол консенсусу SPECTRE та побудова його стійкої модифікації

Виконала : студенка 4 курсу, групи ФІ-62
(шифр групи)
Жук Анна Анатоліївна _____
(прізвище, ім’я, по батькові) (підпис)

Керівник Ковальчук Людмила Василівна д.т.н., проф _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут

Кафедра математичних методів захисту інформації

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки - 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою

М.М.Савчук
_____
(підпис) (ініціали, прізвище)

«___»_____20__ р.

ЗАВДАННЯ
на дипломну роботу студенту

Жук Анна Анатоліївна
(прізвище, ім'я, по батькові)

1. Тема роботи Атака подвійної витрати на протокол конесенсусу SPECTRE та побудова його стійкої модифікації,
керівник роботи Ковальчук Людмила Василівна д.т.н., проф,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 2020 р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи визначення особливостей поведінки постачальника для запобігання атаки подвійної витрати та на основі цього, побудова модифікації протоколу консенсусу SPECTRE

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Визначення напрямку дослідження	1.09-1.10	
2.	Опрацювання матеріалу	1.10-1.02	
3.	Узгодження теми дослідження	1.02-1.03	
4.	Аналіз протоколу консенсусу SPECTRE	1.03-15.03	
5.	Аналіз гібридної атаки на протокол	16.03-2.04	
6.	Побудова модифікації протоколу	3.04-1.05	
7.	Оцінка ймовірності атаки	2.05-22.05	
8.	Формулювання результатів дослідження	23.05-26.05	
9.	Оформлення роботи	26.05-04.06	

Студент _____ Жук А.А.
(підпис) (ініціали, прізвище)

Керівник роботи _____ Ковальчук Л.В.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Кваліфікаційна робота містить: 49 стор., 3 рисунки, 2 таблиці, 16 джерел.

У даній роботі проаналізовано та описано гібридну атаку подвійної витрати на протокол консенсусу SPECTRE. Було описано процедуру вирішення колізії між двома кофліктуючими блоками. Також, було побудовано модифікацію протоколу консенсусу SPECTRE, яка стійкою до гібридної атаки подвійної витрати та оцінено ймовірність такої атаки.

Метою дослідження є аналіз можливих напрямків для побудови модифікації протоколу консенсусу SPECTRE, яка буде стійкою до гібридної атаки подвійної витрати.

Об'єктом дослідження є процес функціонування протоколів консенсусу на базі блокграфу.

Предметом дослідження виступає побудова оцінок стійкості блокграфу до атаки подвійної витрати.

ГРАФЧЕЙН, ПРОТОКОЛ КОНСЕНСУСУ SPECTRE, БЛОКЧЕЙН, АТАКА ПОДВІЙНОЇ ВИТРАТИ

ABSTRACT

The qualifying paper contains: 49 pages, 3 figures, 2 tables, 16 sources.

This paper analyzes and describes a hybrid double-spend attack on the SPECTRE consensus protocol. The procedure for resolving the collision between two conflicting blocks was described. Also, a modification of the SPECTRE consensus protocol was built, which is resistant to a hybrid double-spend attack, and the probability of such an attack was estimated.

The purpose of the paper is to analyze possible directions for constructing a modification of the SPECTRE consensus protocol that will be resistant to a hybrid double-spend attack.

The object of the research is the process of functioning of consensus protocols on the basis of the graphchains.

The subject of the research is the construction of estimates of the resistance of the graphchains to the double-spend attack.

GRAPHCHAIN, SPECTRE CONSENSUS PROTOCOL, BLOCKCHAIN, DOUBLE-SPEND ATTACK

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PoW — Proof-of-Work, доказ виконаної роботи

DAG — спрямований ациклічний граф

PoS — Proof-of-Stake, підтвердження частки

SHA256 — одностороння хеш-функція

ВСТУП

Актуальність дослідження.

З моменту створення блокчейн технологій пройшло вже достатньо часу, щоб зрозуміти, що вони є не такими ж і надійними, незважаючи на те, що щодня вони стають все більш поширеними. Звісно, у них є ряд своїх переваг над класичними технологіями, але нажаль недоліки у них також присутні. До переліку основних дефектів входить назька здатність до масштабування. Стійкість блокчейну до атак істотно знижує збільшення розміру блоку чи навіть підняття швидкості з якою додаються блоки у ланцюг.

Було багато спроб підвищити пропускну здатність криптовалют на основі блокчейну, але вони не були успішними. Це і змусило дослідників розглядати нові способи зберігання децентралізованих транзакцій. Найдалекосяжнішою з них, є технологія узагальнення блокчейну, у якій блоки з транзакціями зберігаються у структурі даних, що має вигляд дерева - графчейн.

Провівши огляд основних проблем протоколів консенсусу побудованих на блокграфах, було визначенно, що певні модифікації зможуть задовільнити потреби користувачів, які стосуються безпечного використання. Одним з вже відомих протоколів консенсусу побудованих на блокграфах є консенсус протокол SPECTRE. Розробники даного протоколу обіцяють достатньо велику пропускну здатність порівняно з протокол консенсусу Накамото, та при цьому можуть гарантувати безпеку всім користувачам. Але нажаль, стійкість цього протоколу до атак зловмисників не підтверджена строгими математичними обгрунтуваннями, незважаючи на гучні заяви авторів.

Тому, модифікація протоколу консенсусу SPECTRE, яка буде стійкою до атаки подвійної витрати є досить актуальною темою для дослідження.

Метою дослідження є аналіз можливих напрямків для побудови

модифікації протоколу консенсусу SPECTRE, яка буде стійкою до гібридної атаки подвійної витрати. **Задача дослідження** полягає у визначенні,за яких умов, ймовірність виконання атаки подвійної витрати на протокол консенсусу SPECTRE буде незначною.

Досягнення поставленої мети передбачає виконання таких **завданнь дослідження**:

1) аналіз основних проблем протоколів консенсусу побудованих на блокграфах;

2) детальний аналіз протоколу консенсусу SPECTRE;

3) опис та аналіз гібридної атаки подвійної витрати на даний протокол;

4) аналіз процедури вирішення конфлікту між двома конфліктуючими транзакціями;

5) аналіз поведінки постачальника для запобігання атаки подвійної витрати.

Об'єктом дослідження є процес функціонування протоколів консенсусу на базі блокграфу.

Предметом дослідження виступає побудова оцінок стійкості блокграфу до атаки подвійної витрати.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: теорія ймовірності та комбінаторика.

Наукова новизна отриманих результатів: вперше показано, як за допомогою зміни поведінки постачальника, можна запобігти атаці подвійної витрати на протокол консенсусу SPECTRE.

Практичне значення. Отримані результати є корисними як для постачальника так і, для зловмисника. Для постачальника, практична значимість полягає у тому, що він може бути впевненим у збереженні коштів, які йому передає покупець. А з точки зору зловмисника, дані результати можуть допомогти у створенні точнішої стратегії атаки на блоки з транзакціями.

1 ОГЛЯД ПРОТОКОЛІВ КОНСЕНСУСУ

У цьому розділі ми розглянемо новітні криптовалюти, що забезпечують децентралізований обіг платежів з використанням публічного журналу транзакцій, який не потребує залучення довіреної особи. Журнал транзакцій, що є результатом роботи усіх учасників децентралізованої системи, створюється на основі протоколу консенсусу, що в свою чергу є захищеним від дій нечесних майнерів з обмеженим ресурсом.

1.1 Огляд основних проблем, для вирішення яких потрібна модифікація протоколу консенсусу

Раніше запропонований Накамото протокол, використовує консенсус PoW [1] та має низьку пропускну здатність. Середня швидкість опрацювання транзакцій у мережі Bitcoin – близько 7 платежів за секунду [2]. Деякі інші децентралізовані криптовалюти мають більші значення, але все ж залишаються малими у зіставленні з централізованим фінансовим обслуговуванням. Це через те, що обмеження, запропоновані Накамото, ставляться на розмір блоку та на інтенсивність генерації блоків. Зміна цих параметрів призводить до часткової централізації або суттєвого зниження стійкості до атаки подвійної витрати, або атаки розгалуження. Для прикладу, якщо збільшується розмір блоку, то і час затримки блоку теж зростає, ці фактори підвищують ймовірність атаки подвійної витрати та атаки розгалуження. Зменшення часу виходу блока - скорочує кількість роботи для створення блоку, що може призвести до збільшення ймовірності ненавмисного форку; до збільшення ймовірності атак – за наявності добре синхронізованого зловмисника. У даний час вирішення проблеми розгалуження здійснюється згідно протоколу

Накамото, за правилом довшого ланцюга. Якщо змінити хоча б один з названих параметрів, то такий спосіб уже не буде захищати мережу. Це означає, що за збільшення розміру блоку або частоти виходу блоків, частина обчислювальної потужності чесних майнерів «втрачається», а умовна доля обчислювальної потужності зловмисника зростає. І тоді виникає ситуація, що для успішної атаки з ймовірністю 1, зловмиснику достатньо мати не 50 % усіх потужностей, а значно менше. Для прикладу, у мережі з достатньо великим часом затримки та високою частотою виходу блоків, атака подвійної витрати можлива за 30 % сумарної обчислювальної потужності(хешрейту) зловмисника. Тому зараз є актуальним вирішення питання про модифікацію протоколу консенсусу, що могла б дозволити більш швидку обробку транзакцій зі збереженням стійкості до основних атак.

Розглянемо всі проблеми, які існують на сьогоднішній день у мережі блокчейн, що дослідники хотіли вирішити у запропонованих ними протоколах консенсусу на блокрафі.

1. Швидка обробка транзакцій. Може досягатися за рахунок збільшення об'єму блоків або збільшення інтенсивності їхнього виходу.

2. Збільшення реальної децентралізації. Для кожного протоколу існує свій ресурс необхідний для створення блоків. Тому єдиним шляхом вирішення цієї проблеми є зменшення об'єму цього ресурсу. Для прикладу, ресурсом для протоколу PoW є обчислювальна потужність, для PoS - депозит у відповідній криптовалюті.

3. Зменшення часу очікування підтвердження транзакції. Наприклад, у мережі Bitcoin, блок з транзакцією вважається підтвердженим, якщо після нього було випущено не менше 6 блоків.

4. Лінійний порядок блоків. Необхідний для коректності процедури голосування за контрактами.

5. Збереження стійкості блокграфу до основних типів атак. В блокчейні існують різні способи доведення стійкості для кожної з основних атак: атаки подвійної витрати, атаки розгалуження, атаки

відмови у обробці транзакції або блока, атаки на зміну лінійного порядку (для блокчейну це просто частковий випадок атаки подвійної витрати). Для атаки подвійної витрати було отримано аналітичні верхні оцінки, для деяких інших - лише асимптотичні оцінки у припущенні, що час є неперервною величиною. Тому надзвичайно трудомісткою задачею є побудова лінійного порядку на блокграфі, якій буде стійким до атак.

1.2 Огляд основних протоколів консенсусу на блокграфах

Однією з найперших робіт [3], у якій було наведено узагальнення блокчейну, є робота , в якій було запропоновано структуру під назвою GHOST. В цій роботі замість лінійної структури, введеної Накамото, було побудовано іншу, більш загальну структуру, що являла собою "дерево" блоків.

На відміну від протоколу Накамото, коли кожен блок має лише одного "нащадка", який на нього посилається, структура GHOST допускає існування, взагалі кажучи, довільної кількості таких нащадків. Іншими словами, ця структура допускає існування форків. Форки можуть виникати з двох причин: поганої синхронізації мережі або під час атаки зловмисника. Два блока, побудованих на одній "висоті", можуть обидва бути валідними. Кожен лінійний ланцюжок цього дерева, історія транзакцій якого узгоджена, є валідним.

Такий підхід до консенсусу дозволяє збільшити пропускну здатність протоколу, хоча все одно деяка кількість створених блоків може "втратитись". Згідно протоколу, кожен майнер, при наявності кількох ланцюжків, може обирати не той, який йому видається більш "правильним". У випадку виникнення конфлікту між блоками валідним вважається той блок, який має більшу кількість "нащадків" (враховуючи форки).

Незважаючи на переваги описані в протоколі авторами, проте у їх роботах немає вичерпних обґрунтувань його стійкості до основних атак, а

також існують суттєві математичні помилки.

Дві наступні роботи тих самих авторів і за цією ж тематикою мають назви SPECTRE [4] та PHANTOM [5] . У них також просувається ідея побудови графчейну, причому запропоновані у роботах протоколи суттєво відрізняються між собою. Єдина спільна риса цих робіт – наявність грубих математичних помилок та відсутність строгих математичних обґрунтувань до своїх гучних заяв.

Крім описаних вище робіт, цікавими є також результати інших авторів, опубліковані у роботах Graphchain [6], Tangle [7], та деякі інші. Всі вони пропонують використовувати блокграф як узагальнення блокчейну. Основною метою таких робіт є збільшення потужності мережі за умови збереження її стійкості до атак. Ці протоколи дійсно покращують характеристики протоколу Накамото, такі як висока пропускна спроможність, час підтвердження блоку, збереження лінійного порядку, тощо.

Найпершою роботою, в якій описується криптовалюта, що базується на графчейні (у ній використовується термін DAG), є блог-пост 2012 року. Ця робота описує криптовалюту DagCoin [8], побудовану без блоків, яка була презентована у 2015 році. Роль блоків тут відіграють окремі транзакції. Такий підхід не виявив якихось суттєвих переваг, проте започаткував новий тип протоколів консенсусу, які базуються на конструкції DAG.

Найпершою реально існуючою криптовалютою, яка використовує DAG замість блокчейну, є криптовалюта IOTA [7] . Протокол консенсусу, що використовується для цієї валюти, називається Tangle (павутина). Згідно протоколу майнінга, кожен наступний блок посилається на два попередніх. Для розв'язання конфліктної ситуації при наявності суперечливих транзакцій пропонується обчислити "висоту" конфліктуючих блоків та вибрати той, який знаходиться на більшій висоті. Такий підхід дає чудові можливості для атаки подвійної витрати, про що розробникам протоколу було повідомлено у приватному

листуванні. Більш того, на конференції CryBlock-19 один з доповідачів по протоколу Tangle підтвердив, що на даний час не існує математично обґрунтованого доведення стійкості цього протоколу до атак, зокрема до атаки подвійної витрати. Тому виникає враження, що цей протокол не є стійким, просто зловмисники ще не виявили достатньої зацікавленості в атаках на нього.

1.3 Протоколи з контролем доступу

Hedera Hashgraph (Хешграф) [9] є протоколом консенсусу, що базується на блокграфі. Цей протокол призначений для використання в середовищах з контролем доступу. Hashgraph є повністю асинхронним протоколом, що означає наявність властивостей стійкості та живучості без якихось додаткових допущень на час синхронізації. Hashgraph гарантує повне вирішення проблеми Візантійської угоди за таких умов: не більше, ніж 1/3 ресурсів контролюється зловмисниками; зловмисник може видаляти повідомлення між чесними учасниками або затримувати їх на необмежений час. У той же час, для досягнення стійкості необхідно, щоб не менше за 2/3 від усіх учасників повинні весь час знаходитись в мережі; всі ці учасники мають бути чесними. Обробка транзакцій базується на неперервній синхронізації станів учасників при отриманні нової інформації щодо транзакцій або нової інформації про оновлення деякими учасниками інформації (“gossip about gossip” – “плітки про плітки”;). Hashgraph має надзвичайно високу пропускну здатність (близько 250 000 трансакцій в секунду [10]), але слід пам’ятати, що він діє лише у середовищі з контролем доступу. Така висока пропускна здатність є можливою завдяки відсутності значного перевантаження мережі: близько nlogn для того, щоб транзакцію ”побачили” всі учасники мережі.

Слабкістю цього протоколу є те, що він орієнтований на середовища з контролем доступу. Тому, неможливо використати цей протокол у

системах, які вимагають повної децентралізації.

Нещодавно представлений протокол Casanova також орієнтований на використання у середовищах з контролем доступу. Подібно до більшості таких протоколів, Casanova є частково синхронним, що потребує гарантій його живучості. Однією з важливих характеристик цього протоколу є підтримка часткового порядку транзакцій замість повного, або лінійного, порядку. Тобто у цьому протоколі транзакції не є повністю впорядкованими. В протоколі Casanova блоки створюються валидаторами через регулярні часові інтервали. Консенсус досягається за кілька раундів. Він базується на голосуванні, яке виконують валідатори. Для оптимізації процесу досягнення консенсусу, Casanova підтримує кілька протоколів, використання яких залежить від ситуації. Тобто у різних умовах використовуються різні протоколи, як при умові наявності конфліктних трансакцій, так і за умови їх відсутності. На сьогоднішній день, не існує інформації щодо практичного застосування протоколу Casanova.

1.4 Протоколи без контроля доступу

Протокол SPECTRE [4]був презентований у 2016 році. У цьому протоколі новостворений блок одразу розсилається. Кожен новий блок посилається на всі валідні блоки у графі, які він бачить на момент створення, і на які ще немає посилань. За наявності колізії протокол передбачає процедуру, що нагадує “голосування”. У даній процедурі голосують не учасники, а самі блоки. Кожен блок у блокграфі голосує за один із блоків у яких конфліктують транзакції. Легітимним вважається той блок за який проголосувала більше блоків. Положення блоку у графчейні визначає як саме цей блок повинен голосувати. Така процедура голосування у SPECTRE забезпечує швидке підтвердження блоків та добре масштабування. Зловмиснику дана можливість підтримувати рівновагу між двома конфліктуючими транзакціями, це говорить про те,

що протокол вразливий до узагальненої атаки розгалуження. Крім цього, процедура голосування у SPECTRE прописана нечітко, що допускає його неоднозначне трактування. Зокрема, це дає можливість для реалізації атаки подвійної витрати, при виконанні певних додаткових умов, які є цілком ймовірними.Робота, у якій презентується цей протокол, також не містить математично обґрунтованих тверджень стосовно оцінок ймовірностей основних атак на блокчейн. Замість них наводяться певні емпіричні міркування, які аж ніяк не можна вважати доведеннями.

У 2018 автори протоколу SPECTRE запропонували новий, зовсім інший протокол – так званий блокграф PHANTOM [5] . Як стверджують автори, цей протокол розв'язує багато з зазначених вище проблем блокграфу. Він зовсім не схожий на протоколи GHOST та SPECTRE. Замість роботи з окремими блоками або їх парами, новий протокол працює з так званими кластерами – множинами блоків, в яких блоки сильно пов'язані між собою. Інші блоки, навпаки, мають мало зв'язків з блоками, які належать кластеру. Вважається, що блоки, які створюються чесними майнерами, будуть сильно пов'язаними з іншими блоками, які теж створені чесними майнерами. Тому припускається, що найбільший кластер, з переважаючою імовірністю, буде містити лише блоки, створені чесними майнерами. Також наводиться протокол побудови лінійного порядку, який суттєво залежить від вибраного основного кластера. Протокол консенсусу полягає у пошуку найбільшого (або майже найбільшого) кластера. Протокол є досить складним та суперечливим. Оскільки він також не описаний чітко та допускає двозначне тлумачення, то можна змоделювати ситуацію, при якій виконання цього протоколу призведе до зациклення. Або при якій протокол не включить до кластеру блок, який має сильні зв'язки з іншими блоками кластера. Слід зазначити, що задача знаходження правильного кластеру є NP-повною, тому у реалізації протоколу пропонується використовувати лише деякий спрощений розв'язок задачі. Це й призводить до некоректності роботи протоколу. Автори PHANTOM проголошують високу пропускну

здатність протоколу та наявність лінійного порядку, проте час підтвердження транзакції, а також час стабілізації блокграфу є дуже довгим. Вважається, що блок є стабільним тоді, коли на нього посилається деякий блок спеціального вигляду, що має назву “пісчаний годинник”. Цей блок має бути таким, що посилається на всі блоки основного кластеру. Автори намагались оцінити імовірність появи такого блоку та, відповідно, середнього часу до його очікування. Для цього вони застосовували апарат теорії ймовірності, проте з величезними математичними помилками. Проте їм все одно не вдалось отримати якісь конкретні оцінки часу очікування такого блоку, крім того, що він є скінченним, оскільки імовірність появи такого блоку ненульова.

Для зменшення часу очікування консенсусу автори також пропонують застосувати деякий симбіоз цих двох протоколів – спочатку побудувати основний кластер з використанням протоколу PHANTOM, а потім застосувати протокол SPECTRE для встановлення лінійного порядку на блоках цього кластеру. Ця пропозиція ще недостатньо розроблена і зараз лише досліджується .

Протокол Graphchain, запропонований Boyen, обіцяє не тільки вирішення питання масштабування, але й зменшення олігополії великих майнінгових пулів. На відміну від протоколів SPECTRE та PHANTOM, Graphchain працює відразу з окремими транзакціями, не об’єднуючи їх у блоки. Кожна транзакція посилається на вибрані батьківські транзакції, тобто ті, які є валідними і були створені та оброблені раніше. Для обробки кожної транзакції потрібно виконати деяку роботу, тобто цей протокол є PoW-подібним. Алгоритм вирішення конфліктної ситуації між транзакціями подібний до того, що реалізований у Bitcoin. Для кожної з конфліктних транзакцій обчислюється її висота – величина, що дорівнює сумарній роботі (PoW) цієї транзакції та всіх інших, на яких ця транзакція посилається (безпосередньо чи опосередковано), тобто транзакцій, що є її предками. Транзакція, що має найбільшу висоту, вважається валідною, а відповідна конфліктуюча транзакція знищується.

Автори протоколу стверджують, що такий спосіб підтвердження транзакцій спонукає майнерів не конфліктувати, а співпрацювати. Крім того, відносно невеликий об'єм PoW дозволяє зменшити олігополію майнінгових пулів. На відміну від Bitcoin, обробка кожної окремої транзакції в Graphchain винагороджується: верифікація транзакції вважається корисною працею, яка має бути оплачена.

Протокол Prism [11] поділяє блоки на кілька класів, згідно до їх функцій: пропозитори, воутери (тобто ті, що голосують) та блоки, що обробляють транзакції. Блоки-пропозитори посилаються на блоки з транзакціями, тим самим підтверджуючи ці блоки. Блоки-воутери вирішують, які саме з блоків- пропозиторів будуть відібрані для формування основного ланцюга. Цей основний ланцюжок і буде визначати стан журналу транзакцій – як ланцюг блоків у Bitcoin. Блоки-воутери об'єднуються у кілька різних ланцюгів (в залежності від значення хеш у цьому блоці), які функціонують паралельно. Така організація блоків дозволяє масштабувати різні функції блокчейну (обробка транзакцій, підтвердження і тд) паралельно. Автори стверджують, що пропускна здатність цього протоколу та його латентність можуть масштабуватись аж до фізичних обмежень на мережу, зі збереженням стійкості до основних атак.

Протокол Fruit Chain [12] теж розділяє блоки за функціями. Частина блоків, з великим значенням PoW, формують основний ланцюжок, як у Bitcoin. Крім цих блоків, існують також блоки з суттєво меншим значенням PoW, які використовуються тільки для обробки транзакцій. Ці блоки називаються "фрукти". Блоки з основного ланцюжка підтверджують блоки-фрукти.

Стійкість такого протоколу можна доводити за тими ж принципами, що й для Bitcoin – достатньо довести відповідні твердження для основного ланцюжка. Одна з найсуттєвіших переваг Fruit Chain – малий час очікування до обробки транзакцій, за рахунок великої кількості маленьких блоків-фруктів. Проте цей протокол ніяк не знижує

час очікування до підтвердження транзакції, тобто після її першого підтвердження блоком з основного ланцюжка все одно потрібно чекати, поки вийде певна кількість блоків з "великим" PoW.

Протокол Parallel Chains [13] має характеристики, аналогічні до характеристик протоколів Fruit Chain та Prism. Однією з основних його переваг є те, що він може застосовуватись і для PoW, і для PoS. Згідно до протоколу, блоки організовані у декілька (від кількох десятків до кількох сотень) практично незалежних ланцюжків. Перший ланцюжок є синхронізаційним. Кожен його блок посилається тільки на попередній блок цього ж ланцюжка. Блоки іншого ланцюжка посилаються на попередній блок цього самого ланцюжка і на останній блок першого ланцюжка, який вони бачать на момент створення. Лінійний порядок блоків у цьому протоколі забезпечується саме завдяки наявності синхнізаційного ланцюжка. Причому завдяки деяким додатковим деталям протоколу транзакції у блоках різних ланцюжків не перетинаються, внаслідок чого швидкість обробки транзакцій зростає прямо пропорційно кількості ланцюжків. Такий протокол дійсно забезпечує суттєво вищу пропускну спроможність, а обґрунтування його стійкості до основних атак є таким самим, як і в традиційному блокчейні – для кожного ланцюжка окремо. Проте, знову-таки, цей протокол аж ніяк не дає можливості зменшити період очікування стабілізації графу, зокрема період очікування повного підтвердження транзакції.

Висновки до розділу 1

У даному розділі було наведено опис технології графчейн, що є узагальненням технології блокчейну. Також було зазначено, які переваги й недоліки має така конструкція. Описано значну кількість протоколів консенсусу, які пропонуються для графчейну. Відмічено особливості нових конструкцій, їх сильні сторони та особливості, які можуть спричинити потенційні вразливості до певних типів атак на блокграф.

Було показано, що жоден з них не має на даний час доведеної стійкості до класичних атак, наприклад, до атаки подвійної витрати. На даний момент розроблено новий протокол консенсусу на блокграфі, який у випадку конфліктних трансакцій є повним аналогом правила довшого ланцюжка. Для такого протоколу повністю обґрунтована стійкість до атаки подвійної витрати, проте стійкість до інших атак ще не досліджувалась.

2 ПРОТОКОЛ КОНСЕНСУСУ SPECTRE

У даному розділі буде розглянуто протокол консенсусу SPECTRE, що був запропонований авторами, як альтернатива протоколу консенсусу Накамото, який дає великі можливості для масштабування.

У SPECTRE кожен блок проіндексований і занесений у журнал транзакцій. Технічно, SPECTRE, зводить блокчейн Накамото до стану спрямованого ациклічного графу (далі "DAG"). Зберігаючи повний DAG блоків, SPECTRE дозволяє майнерам створювати конкурентні блоки і робити це частіше. Це рішення у проектуванні прийняли з метою уникнути необхідність вузлам мережі узгоджувати своє різне бачення ланцюга, щоб зменшити витрати часу на підтвердження ідентичності ланцюгів при створенні нового блоку.

Основна методика SPECTER - алгоритм голосування, щодо порядку між кожною парою блоків у DAG. Виборці - це блоки (а не майнери); голосування кожного блоку інтерпретується алгоритмічно (і не надається інтерактивно) відповідно до його місця в DAG. Сукупний голос більшості дуже швидко стає незворотним, тому більшість голосів використовується для отримання послідовного набору транзакцій.

2.1 Набір інструментів для оцінки безпеки та масштабованості протоколів криптовалют

У фреймворку SPECTRE, протокол криптовалюти складається з двох наборів правил: *протокол майнінгу*, що описує створення блоків і формування журналу транзакцій; та *TxO протокол*, що показує як інтерпретувати журнал транзакцій та як дістати з нього послідовну підмножину валідних транзакцій. Оскільки в протоколі описано, що транзакція буде прийнята з більшою вірогідністю з плином часу,

користувачі додатково запускають *протокол Robust TxO*, щоб кількісно оцінити надійність прийнятої транзакції - це обмежує ймовірність того, що вона коли-небудь буде відмінена, якщо зловмисник намагатиметься перехопити її (наприклад, транзакції з біткойнами можуть бути відмінені, якщо зловмиснику вдасться створити довший альтернативний ланцюг, на якому вони відсутні. Ймовірність виконання цього сценарію зменшується з плином часу). Автори представили фреймворк протоколу у абстрактному вигляді для збереження його загальності.

Транзакції. Транзакція зазвичай позначається як tx. $inputs(tx)$ - це послідовність транзакцій, які мають бути прийняті до того моменту як прийметься транзакція tx; це транзакції з яких взяті гроші, які були витрачені в tx. Дві різні транзакції tx_1 та tx_2 конфліктують, якщо вони поділяють один вхід, тобто двічі витрачаються одні й ті самі гроші; тоді записується

$$tx_2 \in conflict(tx_1)$$

(це симетричне відношення).

Протокол майнінгу. Ми позначаємо N набір вузлів, так званих майнерів. Майнери зберігають і розширюють журнал транзакцій, додаванням нових транзакцій та оголошенням повідомлень, згідно з *протоколом майнінгу*. Час поширення повідомлення розміром B для всіх вузлів в системі вважається меншим за $D = D(B)$ секунд. Наразі ми розглядаємо протокол майнінгу як абстрактний набір правил, яких повинні дотримуватися майнери. Ми позначаємо *чесними* набір вузлів, які завжди слідують вказівкам протоколу, і *зловмисниками* - доповнення цього набору. У сімействі протоколів, на яких ми зосереджуємось, майнери володіють обчислювальною потужністю та виконують перевірку роботи (PoW). Позначимо через α відносну обчислювальну потужність зловмисника. Формально є ймовірність того, що творець наступного PoW в системі належить *зловмисникам*; це добре визначено, оскільки створення PoW моделюється як процес без пам'яті [1], [14].

Формування журналу транзакцій. Результатом протоколу майнінгу є абстрактна структура загальнодоступних даних G, що містить транзакції, так званий журнал транзакцій. Вузли копіюють головний журнал транзакцій локально. Оскільки вони можуть мати дещо різні погляди на журнал транзакцій, оскільки блокам потрібен час для розповсюдження на всі вузли, G_t^v це стан журналу, який спостерігається вузлом v у час t; записується G_t, коли локальний контекст є неважливим.

Протокол TxO. Враховуючи загальнодоступний журнал транзакцій G, *протокол* TxO витягує послідовну підмножину транзакцій з G, позначається $TxO(G)$. Кожна транзакція в цьому наборі повинна мати свої входи, і не може суперечити іншій транзакції в наборі.

Протокол Robust TxO. Користувачі системи повинні отримати гарантії щодо своїх платежів. В основному гарантується, що транзакції будуть прийняті всіма користувачами, і що вони залишиться таким назавжди. Маючи G_t, *протокол* $RobustTxO$, що вказує на підмножину $TxO(G_t)$, позначається $RobustTxO(G_t)$, що представляє собою набір прийнятих транзакцій, які гарантовано залишаються такими назавжди, аж до ймовірної помилкиϵ. $RobustTxO$ приймає як вхідний G_t^v (локальна копія v), так і значення D, λ, α та ϵ . Тут вважається, що ці значення відомі.

Бажані властивості. Таким чином, для протоколу криптовалюти важливі наступні властивості:

Властивість 2.1 (Узгодженість). Прийнятий набір послідовний для будь-якого журналу G:

1) якщо $tx \in TxO(G)$ і $tx_2 \in inputs(tx)$ тоді $tx_2 \in TxO(G)$.

2) якщо $tx \in TxO(G)$ і $tx_2 \in conflict(tx)$ тоді $tx_2 \notin TxO(G)$.

Властивість 2.2 (Безпека). Якщо транзакція надійно прийнята вузлом, тоді з великою ймовірністю, вона буде прийнята всіма вузлами назавжди і приблизний час очікування настання такої події - константа.

Формально,$\forall \epsilon > 0$, $\forall v \in N$, якщо

$$tx \in RobustTxO(G_t^v, D, \lambda, \alpha, \epsilon),$$

тоді з вірогідністю щойнайменше $1 - \epsilon$, існує час $\tau \geqslant t$ такий, що

$$\forall u \in N, \forall s \geqslant \tau : tx \in RobustTxO(G_s^u, D, \lambda, \alpha, \epsilon).$$

Якщо така подія настає, то приблизний час очікування $\tau - t$ – константа.

Властивість 2.3 (Слабка живучість). Якщо транзакція опублікована в журналі транзакцій, вона надійно приймається будь-яким вузлом через короткий час, за умови, що його введення в журнал прийнято і що не публікуються суперечливі трансакції. Формально, нехай

$$v \in N, tx \in G_v \epsilon > 0.$$

Позначимо

$$\psi(t, D, \lambda, \alpha, \epsilon) := min\{s \geqslant t : tx \in RobustTxO(G_s^v, D, \lambda, \alpha, \epsilon)\}$$

час очікування прийняття транзакції вузлом v. Тоді,

$$\mathbb{E}[\psi - t | inputs(tx) \subseteq TxO(G_\psi^v) \wedge conflict(tx) \cap G_\psi^v = \varnothing]$$

– константа.

Означення 2.1. *Поріг безпеки* протоколу криптовалюти визначається максимальним α (відносна обчислювальна потужність зловмисника), для якого виконуються властивості 1-3.

Очікувані значення $\tau - t$ та $\psi - t$, записані у властивостях 2 та 3, визначають приблизний час очікування підтвердження транзакцій у даному протоколі.

Під "слабкістю"у Властивості 3 розуміється, що ніхто не гарантує позитивного рішення конфлікту у випадку, якщо конфліктні транзакції будуть опубліковані незабаром одна за одною.

2.2 SPECTRE vs Bitcoin - огляд

SPECTRE використовує багато рішень Bitcoin. Зокрема, майнери створюють блоки, що представляють собою партії транзакцій. Валідний блок повинен містити рішення проблеми PoW (наприклад, Bitcoin використовує PoW, заснований на часткових колізіях SHA256). Швидкість створення блоку, позначена λ, підтримується постійною протоколом за допомогою періодичних коригувань складності PoW. Розмір блоку обмежений деякими B.

Пропускна здатність біткойна може бути збільшена шляхом збільшення або обмеження розміру блоку (що, в свою чергу, збільшує D), або/та швидкості створення блоку λ. На жаль, добре встановлено, що поріг безпеки Консенсусу Накамото погіршується зі збільшенням $D \cdot \lambda$:

Теорема 2.1. *(Біткойн не масштабований) Поріг безпеки протоколу Bitcoin прямує до нуля, коли $D \cdot \lambda$ зростає.*

Щоб підтримувати високий поріг безпеки, Bitcoin зменшує свою пропускну здатність, зберігаючи λ малим - 1/600 блоків в секунду. Цей великий запас безпеки необхідний, оскільки λ (і B) приймаються раз і назавжди на початку створення протоколу. Отже, навіть якщо мережа надійна і D низька, Bitcoin страждає від низької пропускної здатності - від 3 до 7 транзакцій в секунду, і великого часу конфігурації - десятків хвилин. Але навпаки, пропускну здатність SPECTRE можна збільшити без погіршення порогу безпеки:

Теорема 2.2. *(SPECTRE масштабується) Для будь-якого $D \cdot \lambda$ поріг безпеки SPECTRE становить 50%.*

Звичайно, λ не може збільшуватись безкінечно, інакше мережа буде

переповнена повідомленнями (блоками) і стане перевантаженою. Теорема 2.2 "живе"в теоретичному фреймворку, який не моделює обмеження на пропускну здатність вузлів та пропускну спроможність мережі. Практично, ці бар'єри дозволяють пропускати тисячу операцій за секунду.

Асимптотично, час підтвердження SPECTRE – це

$$\Theta(\frac{ln(1/\epsilon)}{\lambda(1-2\alpha)} + \frac{D}{1-2\alpha}).$$

На практиці, це означає, що час підтвердження займає долі секунди. Під час роботи $RobustTxO$ кожен вузол SPECTRE використовує свою верхню межу останнього D у мережі. Ця межа впливає лише на її власну дію - заниження D приведе до передчасного прийняття транзакцій, а переоцінка його затримає прийняття без необхідності (навідміну від лінійного часу). Важливо, що у випадку сбоїв та тривалих затримок у мережі вузол може переключитися у своєму локальному клієнті на більш консервативне обмеження на D, не погоджуючи це з іншими вузлами.

2.3 Протокол SPECTRE

A. Генерація блоків DAG

Як і в Bitcoin, вузли-учасники (тобто майнери) створюють блоки транзакцій, розв'язуючи задачі PoW. Блок визначає своїх прямих попередників, посилаючись на їх ID у своєму заголовку (ідентифікатор блоку отримується, застосувавши хеш, стійкий до колізій, до власного заголовку). Це записується до структури спрямованого ациклічного графа (DAG) блоків (так як блоки можуть посилатись лише на блоки, створені перед ними), що позначаються, як правило, $G = (C, E)$. Тут C позначає блоки, а E - хеш-посилання. Часто пишеться $z \in G$ замість $z \in C$.

$past(z, G) \subset C$, позначає підмножину блоків досяжних (в термінах DAG) із z, і аналогічно $future(z, G) \subset C$ позначає підмножину блоків, з

яких досяжна z. Це блоки, які можливо створені, до і після появи z, відповідно. Зазначмо, що ребро у DAG спрямоване назад у часі, від новішого блоку до старшого. Вузол не сприймає блок як валідний, доки він не отримає повний набір попередніх блоків. Позначається $cone(z, G)$ набір блоків, які DAG відносить до

$$z : cone(z, G) := past(z, G) \cup \{z\} \cup future(z, G)$$

, також позначається $anticone(z)$ як доповнення до $cone(z, G)$. Набір $past(b, G)$ створюється один раз і для всіх однаковий, при створенні самого b (навідміну від $future(z, G)$ і $anticone(z, G)$, які можуть рости по мірі того, як додаються нові блоки до DAG), тому можна записувати $past(b)$ без прив'язки до контексту.

Унікальний блок *genesis* - це блок, створений на початку створення системи, і кожен дійсний блок повинен мати його в своєму попередньому наборі. Крім того, тут використовується поняття гіпотетичного блоку, $virtual(G)$. Цей блок задовільняє $past(virtual(G)) = G$. Хоча його роль є лише методологічною, $virtual(G)$ також може вважатися таким, що представляє наступний блок, який намагається створити вузол, поточним спостережуваним DAG.

G_t^v позначає блок DAG, що спостерігається вузлом $v \in N$ в момент t. Цей DAG представляє історію всіх (валідних) блоків, отриманих вузлом, інстанціюючи абстрактну структуру даних.

Б. Протокол майнінгу

Правила SPECTRE для майнерів надзвичайно прості:

1) Створюючи або отримуючи блок, передайте цей блок всім учасникам.

2) Створюючи блок, вставте у його заголовок список, що містить хеш усіх блоків-листків (блоків із ступенем 0) у DAG, що спостерігається локально.

Зауважмо, що ці інструкції дозволяють майнерам одночасно

працювати незалежно від потенційних конфліктів у вмісті їхніх блоків[4].

B. Протокол TxO

Огляд. Оскільки блок DAG може містити конфліктуючі транзакції, ми повинні надати вузлам метод інтерпретації DAG та витягнути з нього набір прийнятих транзакцій. Робиться це таким чином, щоб було узгоджено всіма вузлами - головне завдання SPECTER.

Топологія блок-DAG G викликає відношення пріоритету між блоками: якщо x досяжний у (тобто $x \in past(y)$), то x передує y, це можна довести. SPECTER розширює це відношення на повне відношення між блоками G, позначене $\prec$. Цей порядок негайно перекладається на порядок транзакцій у $G : tx_1$ передує tx_2, якщо блок, що містить перший, передує тому, що містить останній. Це відношення, в свою чергу, індукує підмножину прийнятих транзакцій: tx приймається, якщо він передує всім конфліктним транзакціям у G. Відношення $\prec$ породжується процедурою парного голосування, яка відбувається незалежно для кожної пари блоків.

Хоча часом можна посилатися на $\prec$ так, ніби він впорядковує блоки, підкреслемо, що $\prec$ не обов'язково є транзитивним відношенням. Можна створити ряд блоків, які передують один одному циклічно(згідно Парадоксу Кондорсе соціального вибору[16]). Відсутність загального лінійного упорядкування блоків насправді є тим, як SPECTRE використовує слабкіші вимоги до згоди в нашій структурі, оскільки лінійний порядок еквівалентний вирішенню проблеми консенсусу [15].

Попарне впорядкування блоків. Базовий шар SPECTRE включає визначення порядку попарно впорядкованих блоків над блоком-DAG. Зафіксовується два блоки $x, y \in G$. Для того, щоб вирішити, чи $x \prec y$ чи $y \prec x$, структура DAG інтерпретується як репрезентативне голосування. Кожен блок $z \in G$ вважається виборцем щодо пари (x, y), і його голос виводиться із структури DAG. Тут представляється голосування у вигляді чисел $\{-1, 0, +1\}$ і позначається z профіль-голосу на всіх парах $vote(z, G)$. $vote_{x,y}(z, G) = -1$ показує, що

передує y ($x \prec y$), $vote_{x,y}(z, G) = +1$ означає, що y передує x, а $vote_{x,y}(z, G) = 0$ означає нічию. Важливо, що $vote(z, G)$ - це асиметричне відношення:

$$vote_{y,x}(z, G) = -vote_{x,y}(z, G).$$

Для спрощення, також пов'язується голосування з $virtual(G)$. Нагадаємо, що віртуальний блок G - це гіпотетичний блок, який задовольняє

$$past(virtual(G)) = G.$$

Голосування за віртуальний $virtual(G)$ являє собою по суті сукупне голосування всього блок- DAG (довжина всього графу). Основні правила z-голосування для будь-якого $z \in G \cup virtual(G)$ такі:

1) якщо $z \in G$ є в $future(x)$, але не в $future(y)$, то він буде голосувати за x (тобто за $x \prec y$).

2) якщо $z \in G$ - це $future(x) \cap future(y)$, тоz-голосування визначатиметься рекурсивно відповідно до DAG, який зводиться до його минулого, тобто він має такий же голос, що і $virtual(past(z))$. Якщо результатом цього голосування є нічия, z розриває його довільно(в даному випадку "довільно" означає читання заголовку z і виконання правил розв'язання нічиїх, наприклад лексикографічний порядок хешей або інші правила).

3) якщо $z \in G$ не буде в майбутньому жодного з блоків, то він буде голосувати так само, як і голосували більшість блоків у власному майбутньому.

4) якщо z - віртуальний блок G, то він буде голосувати так само, як і голосували більшість блоків у G.

5) остаточно, (у випадку, коли z це x або y), z голосує за те, щоб наслідувати будь-який блок в $past(z)$ і передувати будь-якому блоку, що не належить $past(z)$[4].

Перше правило говорить, що блок, який був чесно опублікований, перемагає у голосуванні ті блоки, які таємно утримуються, оскільки чесні

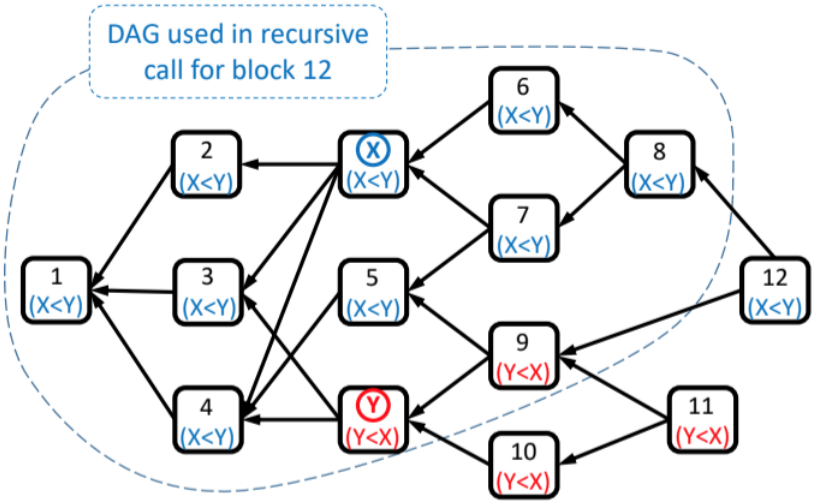

Рисунок 2.1 – Приклад процедури голосування у простому DAG. Блок x і блоки $6-8$ голосують за $x \prec y$, оскільки вони бачать лише x у своєму $past()$, а не y. Аналогічно, блок y і блоки $9-11$ голосують $y \prec x$. Блок 12 голосує відповідно до рекурсивного виклику процедури голосування від DAG з якого виключили блоки $10, 11, 12$. Будь-який блок з $1-5$ голосує $x \prec y$, тому що він бачить більше $x \prec y$ виборців у своєму $future()$, ніж $y \prec x$ виборців.

вузли продовжують додавати нові блоки до свого майбутнього набору блоків. Друге і четверте правила разом гарантують посилення більшості голосів, оскільки нові блоки додають голоси, які підтверджують попередні рішення. Третє правило - найдосконаліше; в основному, воно дозволяє блокам у $past(x)$ (на додаток до тих, у $future(x)$) голосувати за x проти у випадку, якщо y тривалий час приховувались. Це потрібно для протидії атаці пре-майнінгу. Зауважмо, що всі голоси зважають на топологію DAG: Якщо x досяжний з y, то всі блоки голосують за те, що передує .

Властивість 2.4 Якщо блок опублікований, то множина блоків що передують йому (в порядку попарного голосування) дуже швидко обмежується, тобто цьому блокові будуть передувати блоки, що вже були

присутні в DAG, або були додані майже одночасно з даним блоком(але трохи пізніше).

Наслідком цього є гарантії безпеки транзакцій, принаймні, на інтуїтивно зрозумілому рівні: користувач, транзакція якого вбудована у якийсь опублікований блок x, може гарантувати його безпеку, почекавши деякий час після публікації x. Потім йому гарантується, що будь-якому блоку, опублікованому пізніше і який може містити суперечливу транзакцію - передуватиме x, отже, це не повинно загрожувати прийняттю його транзакції.

Прийняття транзакцій. Ознайомившись з попарним впорядкування блоків, перейдемо до розгляду плану прийняття транзакцій. Щоб зберегти послідовність, транзакція позначається як прийнята, якщо виконуються три нижче наведені умови:

1) всі її входи були прийнятими.

2) усі конфліктуючі транзакції, що не пов'язані з даною транзакцією топологічно, містяться в блоках, яким передує блок з даною транзакцією.

3) всі суперечливі транзакції з її минулого набору (тобто ті, які передують їй в DAG, топологічно) були відхилені.

Висновки до розділу 2

У даному розділі представлено протокол консенсусу SPECTRE, новий криптовалютний протокол, який по суті є масштабовним. На відміну від Bitcoin та багатьох його варіантів, автори SPECTRE стверджують, що він захищений від зловмисників, що мають менше 50% обчислювальної потужності, навіть коли його пропускна здатність збільшується і затримка поширення стає незначною. Вище представлені результати абстрактно демонструють, що SPECTRE може досягти неймовірно низьких конфірмаційних часів, особливо порівняно з консенсусом Накамото. Ключовим фактором, так званих, досягнень SPECTRE є його готовність відкласти рішення щодо видимо подвійних

витрат. Цей факт також робить його менш придатним для таких систем, як Ethereum, де необхідний тотальний порядок транзакцій. Основний алгоритм SPECTRE - це процедура голосування в парному порядку, яка є нетривіальною.

3 ОЦІНКА ЙМОВІРНОСТІ УСПІХУ АТАКИ ПОДВІЙНОЇ ВИТРАТИ НА МОДИФІКАЦІЮ ПРОТОКОЛУ SPECTRE

У цьому розділі буде описана гібридна атака на протокол консенсусу SPECTRE, яку можна вважати сукупністю double-spend атаки, Censorship атаки та Splitting атаки. Вона буде містити в собі риси всіх зазначених атак. Буде показано, як за певних умов, на базі гібридної атаки можна провести атаку подвійної витрати. Також, буде приведено варіант модифікації протоколу консенсусу SPECTRE, який буде стійким до такої атаки.

3.1 Особливості поведінки постачальника для запобігання атаки подвійної витрати

У традиційному описі, поведінка постачальника за умови атаки подвійної витрати відбувається наступним чином. Зловмисник робить деяку транзакцію в деякому блоці X_n , та передає ці кошти постачальнику за деякий товар або послугу. Продавець підключається до мережі та бачить, що на блок з транзакцією вже посилаються z блоків підтвердження, яких є достатньо для необоротності транзакції(значення z визначається за допомогою параметрів мережі). Тоді у цей момент він може відправляти товар покупцеві.

Але у нашому випадку, цей протокол має свої особливості у порівнянні з іншими. Основна особливість полягає в тому, що постачальник товару або послуги, коли підключився до мережі Інтернет і бачить блок з транзакцією, навіть якщо на момент підключення блок мав вже достатню кількість z блоків підтвердження, постачальник всерівно чекає z нових блоків підтвердження, не з моменту обробки транзакції, а з того часу коли він вперше побачив оброблену транзакцію.

Тобто, відмінність у нашій стратегії від стратегії описаної авторами у протоколі , полягає у тому, що ми можемо гарантувати необоротність транзакції з ймовірністю близькою до одиниці, тільки тоді, коли блоки підтвердження були зроблені на очах у постачальника.

3.2 Опис гібридної атаки подвійної витрати

Гібридна атака на протокол консенсусу SPECTRE складається з чотирьох етапів.

Перший етап:

– чесний майнер генерує блоки 1, 2, 3, 4, 5, тобто ланцюг CH_1 довжиною h_1;

– у той же час зловмисник генерує блок з транзакцією X_0 та підтверджує його блоками X_1, X_2 , тобто генерує ланцюг CM_1 довжиною m_1 (але не відкриває цей ланцюг).

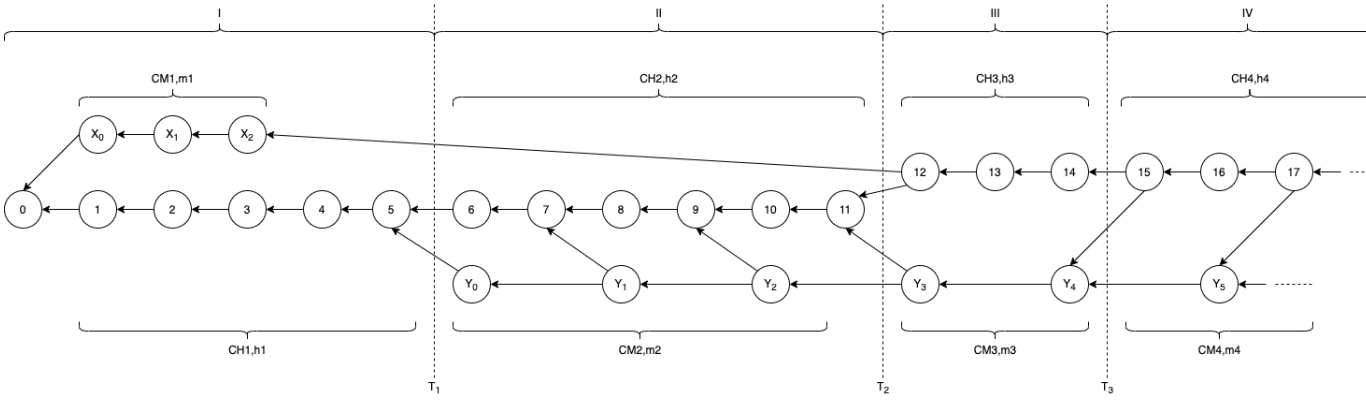

Рисунок 3.1 – Схематичний опис етапів атаки на протокол консенсусу SPECTRE

Другий етап:

Стартуємо з моменту часу T_1, де зловмисник вирішує, що згенерував достатню кількість блоків для підтвердження транзакції X_0.

– зловмисник генерує блок з транзакцією Y_0 та підтверджує його, для цього він генерує ланцюг CM_2 довжиною m_2 (але не відкриває цей

ланцюг);

– чесний майнер генерує ланцюг CH_2 довжиною h_2, який продовжує ланцюг CH_1.

Третій етап:

Стартуємо з моменту часу T_2, де зловмисник вирішує, що згенерував достатню кількість блоків для підтвердження транзакції Y_0.

– зловмисник публікує CM_1;

– продавець чекає деякий час, протягом якого майнери генерують ланцюг CH_3 довжини h_3, а потім надсилає товари чи послуги.

– за цей час зловмисник генерують ланцюг CM_3 довжини m_3, що продовжує підтверджувати блок з транзацією X_0 . Зловмисник все ще не відкриває ланцюги CM_2 і CM_3.

Четвертий етап:

Стартуємо з моменту часу T_3, коли продавець відправляє товари чи послуги.

– у момент часу T_3 продавець відправляє товар;

– відразу після цього зловмисник відкриває два свої ланцюги CM_2, CM_3.

– чесний майнер підтверджує обидва ланцюги CH_3 та CM_3 з ланцюгом CH_4 довжиною h_4;

– зловмисник підтверджує лише ланцюг CM_3 з ланцюгом CH_4 довжиною m_4.

3.3 Процедура вирішення конфлікту між двома конфліктуючими блоками

Процедура вирішення конфлікту відбувається згідно протоколу консенсусу SPECTRE, за допомогою голосування блоків за процедурою описаною у роботі [4], але у нашому випадку блоки голосують наступним чином:

– усі блоки ланцюга CH_3 голосують за X_0 (h_3 блока);

– усі блоки ланцюга CM_3 голосують за Y_0 (m_3 блока);

– усі блоки ланцюга CM_3 голосують за X_0 (m_1 блок);

– усі блоки ланцюгів CM_2, CM_3, CM_4 голосують за Y_0 ($m_2 + m_3 + m_4$ блока);

– усі блоки ланцюга CH_1 та ланцюга CH_2 голосують як більшість у їхньому майбутньому ($h_1 + h_2$ блока);

– усі блоки ланцюга CH_4 голосують як більшість у їхньому минулому (h_4 блока).

На рисунку 3.1 видно, що блок під номером 16 голосує як більшість у минулому, тобто його голосування залежить від того, як проголосував блок під номер 15. Усі наступні створені блоки, після 16, голосують за таким же принципом. Тобто, голосування ланцюга CH_4 повністю залежить від того, як проголосує 15-й блок. Блок під номером 15 - це блок, який перший з усіх блоків "бачить"X_0 та Y_0.

Проналізувавши голосування в $past(15)$ бачимо, що:

– усі блоки з CH_3 голосують за X_0;

– усі блоки з CM_3 голосують за Y_0;

Голосування блоку під номером 11 залежить від того, як співвідносяться довжини згенерованих ланцюгів у момент часу T_2 тобто, чи $h_3 > m3$, чи $h_3 \leq m_3$.

Розглянемо варіант, якщо $h_3 \leq m_3$, то:

– блок 11 голосує за Y_0;

– блок 10 теж голосує за Y_0 та всі попередні блоки підтримують його.

Тобто, один з сценаріїв реалізації успішної атаки можливий за умови, коли у момент часу T_2 довжина ланцюга згенерованого чесним майнером $\leq$ довжини ланцюга згенерованого у той же час зловмисником ($h_3 \leq m_3$).

Для випадку, якщо $h_3 > m_3$ (рахуємо, що $h_3 = m_3 + k$), маємо:

– блок 11 голосує за X_0;

– блок 10 буде голосувати за Y_0 тільки, якщо з моменту генерації 10-го блоку до моменту генерації 11-го блоку, зловмисник згенерує $\geq k + 1$

блок.

Дійсно, за цієї умови в $future(10)$: $h_3 + 1(11 \text{ блок}) \leq m_3 + k + 1$, де $h_3 + 1$ - голосують за X_0 , а $m_3 + k + 1$ - голосують за Y_0.

Твердження 3.1. *(Необхідна умова атаки за умови, що $h_3 > m_3$): $\exists i$ у ланцюгу CH_2 , що між і-м та $i + 1$ блоком, зловмисник згенерував $\geq k + 1$ блоку.*

Якщо ж, умова $h_3 > m_3$ та Твердження 3.1 не виконуються, то ще є можливість виконати атаку в майбутньому. Для цього необхідно, щоб ланцюг зловмисника CM_4 був довший за ланцюг чесного майнера CH_4 . Тобто, на якомусь етапі потрібно, щоб $m_4 \geq h_4 + k$.

Тобто, для успішної атаки необхідне об'єднання усіх нижче наведених випадків:

– Перший випадок (A_1): довжина ланцюга CM_3, що згенерований зловмисником повинна бути $\geq$ за довжину ланцюга CH_3, що знегерований чесним майнером, тобто $m_3 \geq h_3$;

– Другий випадок (A_2): довжина ланцюга CM_3, що згенерований зловмисником $<$ за довжину ланцюга CH_3, що знегерований чесним майнером, тобто $m_3 < h_3$ та виконується Твердження3.1;

– Третій випадок (A_3): довжина ланцюга CM_3, що згенерований зловмисником $<$ за довжину ланцюга CH_3, що знегерований чесним майнером, тобто $m_3 < h_3$ та Твердження3.1 не виконується, але десь у майбутньому довжина згенерованого зловмисником ланцюга перевищить довжиину ланцюга чесного майнера $m_4 > h_4 + k$.

А точніше, ймовірність атаки $\leq P(A_1) + P(A_2) + P(A_3)$.

3.4 Побудова модифікації протоколу консенсусу SPECTRE, яка є стійкою до гібридної атаки подвійної витрати

В основі модифікації лежить зміна поведінки постачальника. Навідмінно від традиційної поведінки постачальника, коли він постачає

товар, як тільки бачить достатню кількість k-блоків підтвердження на блокові у якому йому перераховують кошти , то у нашому випадку, після того, як він вперше побачив блок зі своєю транзакцією , не зважаючи на вже наявну кількість блоків підтвердження, постачальник чекає k-нових блоків підтвердження на своєму блокові . Навіть, якщо на ньому на момент появи вже було багато блоків підтвердження . У даній зміні поведінки і полягає модифікація.

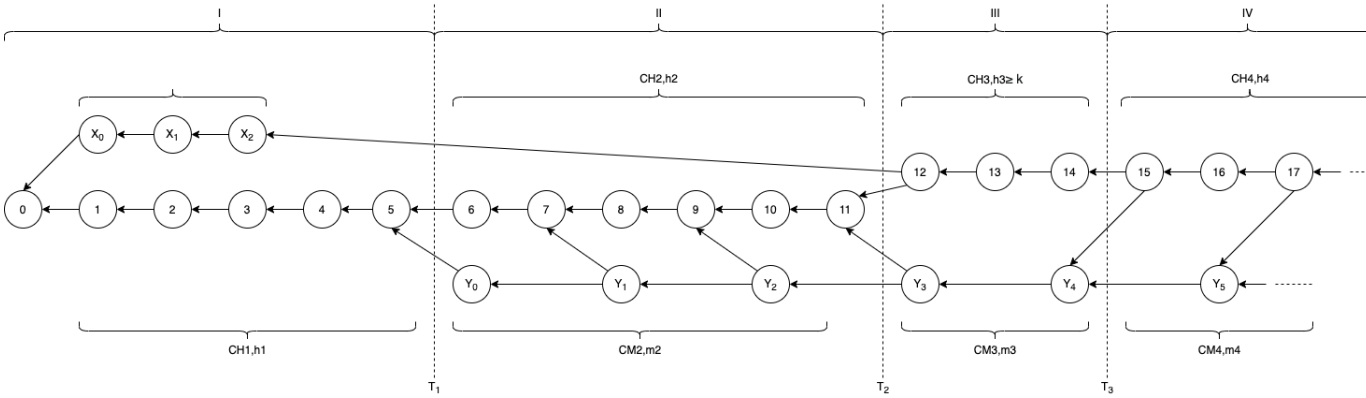

Рисунок 3.2 – Схематичний опис етапів атаки з модифікацією на протокол консенсусу SPECTRE

Перший етап:

– чесний майнер генерує блоки 1, 2, 3, 4, 5, тобто ланцюг CH_1 довжиною h_1;

– у той же час зловмисник генерує блок з транзакцією X_0 та підтверджує його блоками X_1, X_2 , тобто генерує ланцюг CM_1 довжиною m_1 (але не відкриває цей ланцюг).

Другий етап:

Стартуємо з моменту часу T_1, де зловмисник вирішує, що згенерував достатню кількість блоків для підтвердження транзакції X_0.

– зловмисник генерує блок з транзакцією Y_0 та підтверджує його, для цього він генерує ланцюг CM_2 довжиною m_2 (але не відкриває цей ланцюг);

– чесний майнер генерує ланцюг CH_2 довжиною h_2, який продовжує

ланцюг CH_1.

Третій етап:

Стартуємо з моменту часу T_2, де зловмисник вирішує, що згенерував достатню кількість блоків для підтвердження транзакції Y_0.

– зловмисник публікує CM_1;

– продавець чекає деякий час, протягом якого чесні майнери генерують ланцюг CH_3 довжини $h_3 \geq k$, а потім надсилає товари чи послуги (величина k визначається в залежності від параметрів мережі).[1]

– за цей час зловмисник генерують ланцюг CM_3 довжини m_3, що продовжує підтверджувати блок з транзацією X_0 . Зловмисник все ще не відкриває ланцюги CM_2 і CM_3.

Четвертий етап:

Стартуємо з моменту часу T_3, коли продавець відправляє товари чи послуги.

– у момент часу T_3 продавець відправляє товар;

– відразу після цього зловмисник відкриває два свої ланцюги CM_2, CM_3.

– чесний майнер підтверджує обидва ланцюги CH_3 та CM_3 з ланцюгом CH_4 довжиною h_4;

– зловмисник підтверджує лише ланцюг CM_3 з ланцюгом CH_4 довжиною m_4.

3.5 Оцінка ймовірності атаки подвійної витрати

Для оцінки верхньої границя ймовірності успіху атаки подвійної витрати ми будемо використовувати Рисунок 3.1. Хоча у доведенні використовується довільна кількість блоків у кожному з ланцюгів. Також потрібно зазначити, що при зміні будь-якого ланцюга на дерево із такою ж кількістю блоків, нічого не змінюється у моделі атаки та у доведенні

[1]Цей пункт є ключовим для захисту від атаки подвійної витрати. Саме тут поведінка постачалька видрізняється від традиційної. Тому, що у традиційній поведінці, коли постачальник бачить свою транзакцію, і на ній вже є потрібна кількість k - блоків підтвердження, то він одразу надсилає товар.

теореми.

Також нам потрібні описані вище пункти 3.1 та 3.2.

Нехай $\alpha_H, \alpha_M > 0$ - це певні значення, які позначають частку чесних майнерів та зловмисників, відповідно. Ці значення визначаються, як параметри функцій розподілу:

$$F_{T_H} = P(T_H < t) = 1 - e^{-\alpha_H t}, \tag{3.1}$$

$$F_{T_M} = P(T_M < t) = 1 - e^{-\alpha_M t}, \tag{3.2}$$

де T_H, T_M - випадкові величини, які вимірюють час необхідний на майнінг одного блоку для чесного майнера та зловмисника відповідно. Тоді, ймовірність того, що чесний майнер згенерує наступний блок раніше ніж зловмисник становить

$$p_H = \frac{\alpha_H}{\alpha_M + \alpha_H}, p_M = 1 - p_H = \frac{\alpha_M}{\alpha_M + \alpha_H}. \tag{3.3}$$

Також припускаємо, що D_H позначає час необхідний для того, щоб чесний майнер опублікував блок (після генерації) для всіх (принаймі чесних) вузлів мережі; D_M відповідний час публікування блока для зловмисника і тут же припускаємо, що $D_M = 0$, це означає, що противник корумпований і добре синхронізований з мережею.

Також визначимо:

$p_H^{'}$ - ймовірність того, що чесний майнер буде генерувати та поширювати блоки для усіх (принаймі чесних) вузлів раніше за зловмисника;

$p_M^{'}$ - ймовірність настання альтернативної події.

$$p_H^{'} = e^{-\alpha_M D_H} p_H \tag{3.4}$$

$$p_M^{'} = 1 - e^{-\alpha_M D_H} p_H \tag{3.5}$$

Позначимо $P_z(k)$ як ймовірність того, що зловмисники згенерують рівно k

блоків до того, як чесні майнери згенерують та опублікують z блоків. Тоді,

$$P_z(k) = \sum_{i=0}^{k} \left[\binom{z+i-1}{i} p_H^z p_M^k e^{-\alpha_M n D_H} \right] \frac{(\alpha_M n D_H p_H)^{k-i}}{(k-i)!} \tag{3.6}$$

Зауважмо, що у випадку коли $D_H = 0$ ймовірність 3.6 приймає простіший вигляд:

$$P_z(k) = \binom{z+k-1}{k} p_H^z p_M^k \tag{3.7}$$

через те, що сума 3.6 має лише один доданок k : для $k = i$; усі інші дорівнюють нулю через $(\alpha_M n D_H p_H)^{k-i}$.

Теорема 3.1. *Нехай продавець чекає z блоків після того, як він вперше побачив транзакцію X_0. Тоді, з визначень 3.1 - 3.7, верхня границя ймовірності $P_z(\alpha_M, \alpha_H, D)$ успіху зловмисника становить:*

$$\begin{aligned} &P_z(\alpha_M, \alpha_H, D) \leq \\ &\leq 1 - \sum_{k=0}^{z-1} P_z(k) \cdot \left(1 - \left(\frac{p_M^{'}}{p_H^{'}}\right)^{z-k} \left(\left(p_H^{'}\right)^{z-k-1} + 1\right)\right) \end{aligned} \tag{3.8}$$

та у конкретному випадку, коли $D_H = 0$

$$\begin{aligned} &P_z(\alpha_M, \alpha_H, 0) \leq \\ &\leq 1 - \sum_{k=0}^{z-1} p_H^z p_M^k \binom{z+k-1}{k} \times \\ &\times \left(1 - \left(\frac{p_M}{p_H}\right)^{z-k} \left(\left(p_H\right)^{z-k-1} + 1\right)\right) \end{aligned} \tag{3.9}$$

Доведення. Припустимо, що B_1 буде першим блоком, який побачить X_0 та Y_0 у своєму минулому (на Рисунку 3.1 це блок під номером 15). Цей блок та всі блоки у його майбутньому, голосуватимуть ідентично до більшості у своєму минулому. Тому голоси блоків від CH_4 головним чином залежать від голосів блоків у минулому (B_1).

Аналізуючи голоси у минулому (B_1):

– усі блоки з CH_3 голосують за X_0;

– усі блоки з CM_3 голосують за Y_0;

– останній блок CH_2, B_2 (на Рисунку 3.1 B_2 це блок під номером 11) голосує за Y_0 якщо $h_3 \leq m_3$, і так само всі блоки з CH_1 таCH_2 . Таким чином одна з умов успішної атаки це $h_3 \leq m_3$ її ми позначимо як C_1 :

$$h_3 \leq m_3. \qquad (C1)$$

Коли C_1 не виконується, зловмисник ще має шанс на успіх в тому випадку, якщо блоки в $past(B_2)$ проголосують за Y_0. Проаналізуємо, коли це можливо, якщо C_1 не виконується, тоді B_2 голосує за X_0. Нехай $h_3 = m_3 + l$. Тоді блоки що були безпосередньо перед B_2 , будуть названі B_3 (на Рисунку 3.1 це блок номер 10), будуть голосувати за Y_0 тільки в тому випадку, якщо за час між B_3 та B_2 зловмисник згенерує не менше ніж $l + 1$ блок. Аналогічно, якщо B_3 та B_2 голосують за X_0, необхідною умовою щоб попередній блок B_4 проголосував за Y_0 є той факт, що за час між B_4 та B_3 зловмисник згенерував не менше $l + 2$ блоки, і так далі.

Отже, якщо C_1 не виконується, необхідною умовою для настання події “якийсь блок під номером i у ланцюгу CH_2 голосує за Y_0” - це умова C_2:

$\exists B_5 \in CH_2$ такий, що за час між B_5 і наступним блоком зловмисник згенерує не менше ніж $l + 1 + u$ блоків, де u - це кількість блоків ланцюга CH_2, що є в $future(B_5)$.

Якщо C_1 та C_2 не виконуються, зловмисник досі має шанс реалізувати атаку. C_3 є необхідною умовою:

"У якийсь момент $T > T_3$ нерівність буде задовільняти умову: $m_4 \geq h_4 + l$ ".

Тому для успішної атаки хоча б одна з умов C_1, C_2, C_3 повинна бути виконана.

Відповідно до постановки теореми, $h_3 \geq z$. Тоді використовуючи 3.6,

ми отримаємо

$$P(C_1) = 1 - \sum_{k=0}^{z-1} P_z(k). \tag{3.10}$$

Далі,

$$\begin{aligned}
&P(C_1 doesn'thold, C_2 holds) = \\
&= P(C_1 doesn'thold) \times P(C_2 holds) = \\
&= \sum_{k=0}^{z-1} P_z(k)\Big((p_M^{'})^{z-k} + (p_M^{'})^{z-k+1} + ... + (p_M^{'})^{z-k+(k_2-1)}\Big) \leq \\
&\leq \sum_{k=0}^{z-1} P_z(k)\Big(\frac{(p_M^{'})^{z-k}}{1-p_M^{'}}\Big) = \sum_{k=0}^{z-1} P_z(k)\Big(\frac{(p_M^{'})^{z-k}}{p_H^{'}}\Big).
\end{aligned} \tag{3.11}$$

Ймовірність

$$\begin{aligned}
&P(C_1 \ \ and\, C_2 \ \ doesn't\ hold, \ \ C_3\ holds) = \\
&= (1 - P(C_1))(1 - P(C_2)) \cdot P(C_3) \leq \\
&(1 - P(C_1)) \cdot P(C_3).
\end{aligned}$$

Зауважмо, що умова C_3 означає "зловмисник наздоганяє щонайменше $z - k$ позаду тому

$$P(C_3) \leq \Big(\frac{p_M^{'}}{p_H^{'}}\Big)^{z-k}.$$

Тоді,

$$\begin{aligned}
&P(C_1 \ \ and\, C_2 \ \ doesn't\ hold, \ \ C_3\ holds) \leq \\
&\leq \sum_{k=0}^{z-1} P_z(k)\Big(\frac{p_M^{'}}{p_H^{'}}\Big)^{z-k}.
\end{aligned} \tag{3.12}$$

Виходячи з (3.10) - (3.12), ми отримаємо

$$
\begin{aligned}
&P_z(\alpha_M, \alpha_H, D) \leq \\
&\leq 1 - \sum_{k=0}^{z-1} P_z(k) + \sum_{k=0}^{z-1} P_z(k) \frac{\left(p_M^{'}\right)^{z-k}}{p_H^{'}} + \\
&\sum_{k=0}^{z-1} P_z(k) \left(\frac{p_M^{'}}{p_H^{'}}\right)^{z-k} = \\
&= 1 - \sum_{k=0}^{z-1} P_z(k) \Big[1 - \frac{\left(p_M^{'}\right)^{z-k}}{p_H^{'}} - \left(\frac{p_M^{'}}{p_H^{'}}\right)^{z-k}\Big] = \\
&= 1 - \sum_{k=0}^{z-1} P_z(k) \Big[1 - \left(\frac{p_M^{'}}{p_H^{'}}\right)^{z-k} \Big((p_H^{'})^{z-k-1} + 1 \Big)\Big].
\end{aligned}
$$

Теорему доведено

□

У таблицях 1 та 2 наведено мінімальну кількість блоків підтвердження для яких ймовірність нападу не перевищує 10^{-3} для різних параметрів мережі(розрахунок проводиться за допомогою[17]).

Таблиця 3.1 – Мінімальна кількість z блоків підтвердження, для яких $P_z(\alpha_M, \alpha_H, D_H) \leq 10^{-3}$ для різних параметрів мережі $\triangle$ (у секундах) і p_M, і для $\alpha = 0.00167$ (як для BTC)

p_M	D_H			
	0 сек	15 сек	30 сек	60 сек
0.1	6	6	7	7
0.15	9	9	10	10
0.2	14	14	14	15
0.25	21	21	22	24
0.3	33	35	36	40
0.35	60	65	69	81
0.4	137	154	175	228

Таблиця 3.2 – Мінімальна кількість z блоків підтвердження, для яких $P_z(\alpha_M, \alpha_H, D_H) \leq 10^{-3}$ для різних параметрів мережі $\triangle$ (у секундах) і p_M, і для $\alpha = 0.0167$

p_M	D_H			
	0 сек	15 сек	30 сек	60 сек
0.1	6	7	8	11
0.15	9	11	13	20
0.2	14	17	23	43
0.25	21	29	44	172
0.3	33	55	114	$P_z = 1$
0.35	60	139	$P_z = 1$	$P_z = 1$
0.4	137	203	$P_z = 1$	$P_z = 1$

Висновки до розділу 3

У даному розділі було детально розглянуто різницю між поведінкою постачальника у традиційному випадку та поведінкою, яка запобігає атаці подвійної витрати. Було схематично зображено та описано гібридну атаку подвійної витрати.

Також, було представлено, як за допомогою голосування блоків, що відбувається згідно протоколу консенсусу SPECTRE, вирішується проблема між двома конфліктуючими блоками.

Зважаючи на наявність грубих математичних помилок, у описі протолу SPECTRE, для захисту постачальника від атаки подвійної витрати, було побудовано модифікацію протоколу консенсусу SPECTRE, яка є стійкою до гібридної атаки подвійної витрати.

До того ж, посилаючись на всі проведені у цьому розділі дослідження, було оцінено верхню границю успіху атаки подвійної витрати.

ВИСНОВКИ

У даному дослідженні нами було наведено опис технології графчейн. Також було зазначено, які переваги й недоліки має така конструкція. Описавши значну кількість протоколів консенсусу, які пропонуються для графчейну, було відмічено особливості таких протоколів, які можуть спричинити потенційні вразливості до певних типів атак на блокграф. Було показано, що жоден з них не має на даний час доведеної стійкості до класичних атак, наприклад, до атаки подвійної витрати.

Нами було проаналізовано протокол консенсусу SPECTRE, новий криптовалютний протокол, який по суті своїй є масштабовним. Було визначено, що представлені авторами протоколу результати лише абстрактно демонструють те, що SPECTRE може досягти неймовірно низьких конфірмаційних часів, особливо порівняно з консенсусом Накамото.

Також, нами було розглянуто та описано гібридну атаку подвійної витрати на протокол консенсусу SPECTRE. Було проаналізовано поведінку постачальника до моменту передачі товару або послуги. За допомогою цього аналізу було визначено особливості в поведінці,які запобігають атаці подвійної витрати. Також, було побудовано процедуру за допомогою якої можна вирішити конфлікт між двома блоками. З отриманих результатів нами було побудовано модифікацію протоколу консенсусу SPECTRE, яка є стійкою до гібридної атаки подвійної витрати та оцінено ймовірність такої атаки.

Не дивлячись на отримані результати та той факт, що поставлена задача була розв'язана в ході виконання роботи, існує ряд напрямків дослідження, що не були цілковито висвітлені. Оскільки, останніми роками активно розглядається можливість переходу з блокчейну на новішу структуру, а якщо точніше, то на технологію графчейн, яка ще є

не вивчено до кінця. Тому, можливість модифікації протоколів, які побудовані на блокграфі, що будуть стійкими до основних атак, може стати темою для майбутніх досліджень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

2. Bitcoin confirmed transactions per day stats., 2018. [Електронний ресурс]. — Режим доступу: https://www.blockchain.com/charts/n-transactions

3. Yonatan Sompolinsky and Aviv Zohar. Accelerating bitcoin's transaction processing. fast money grows on trees, not chains. IACR Cryptology ePrint Archive, 2013:881., 2013.

4. Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. Spectre: A fast and scalable cryptocurrency protocol. IACR Cryptology ePrint Archive 2016:1159, 2016.

5. Y. Sompolinsky and A. Zohar. Phantom: A scalable blockdag protocol, 2018.

6. Xavier Boyen, Christopher Carr, and Thomas Haines. Graphchain: a blockchain-free scal- able decentralised ledger. In Proceedings of the 2nd ACM Workshop on Blockchains, Cryp- tocurrencies, and Contracts, pages 21–33. ACM, 2018.

7. Serguei Popov. The Tangle (2017)., 2017[Електронний ресурс]. — Режим доступу: https://iota.org/IOTA_Whitepaper.pdf.

8. Sergio Demian. Lerner. dagcoin: a cryptocurrency without blocks, 2015.

9. Leemon Baird. Hashgraph consensus: fair, fast, Byzantine fault tolerance. swirlds tech report, 2016 [Електронний ресурс]. — Режим доступу: https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

10. Yaoqi. Jia. Demystifying hashgraph: Benefits and challenges.[Електронний ресурс]. — Режим доступу: https://hackernoon.com/demystifying-hashgraph-benefits-and-challenges-d605e5c0cee5

11. Vivek Bagaria, Sreeram Kannan, David Tse, Giulia Fanti, and

Pramod Viswanath. De- constructing the blockchain to approach physical limits. arXiv preprint arXiv:1810.08092, 2018.

12. J. Dilley, A. Poelstra, and J. Wilkins. Unfreezable blockchain. bitcoin forum., 2016[Електронний ресурс]. — Режим доступу: https://bitcointalk.org/index.php?topic=57647.msg686497&msg68649

13. Matthias Fitzi, Peter Gaˇzi, Aggelos Kiayias, and Alexander Russell. Parallel chains: Im- proving throughput and latency of blockchain protocols via parallel composition. Cryptology ePrint Archive, Report 2018/1119, 2018[Електронний ресурс]. — Режим доступу: https://eprint.iacr.org/2018/1119

14. Meni Rosenfeld. Analysis of hashrate-based double spending. arXiv preprint arXiv:1402.2009, 2014.

15. Miguel Correia, Nuno Ferreira Neves, and Paulo Ver ıssimo. From consensus to atomic broadcast: Time-free byzantine-resistant protocols without signatures. The Computer Journal, 49(1):82–96, 2006.

16. Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. Handbook of Social Choice & Welfare, volume 2. Elsevier, 2010.

17. Yonatan Sompolinsky and Aviv Zohar. 2018. Phantom. IACR Cryptology ePrint Archive, Report 2018/104 (2018).